

## **AMENDMENTS TO THE CLAIMS**

Prior to the present communication, claims 1, 2, 4, 13, 14 and 16 were pending in the subject application. Each of claims 1 and 13 has been amended herein, while claims 2 and 14 have been canceled. Thus, claims 1, 4, 13, and 16 remain pending. It is respectfully submitted that no new matter has been added by way of the present amendments. All claims currently pending and under consideration in the present application are shown below. This listing of claims will replace all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

1. (Currently Amended) In a threaded computing environment having a plurality of contexts, each context capable of containing a queue, context settings, a context dictionary, and objects, a method for allocating the access of threads to a user interface context, the method comprising:

receiving a request to access the user interface context from a first thread, wherein the user interface context is configured to receive input from a user, to provide output to the user, and to maintain the context settings and the context dictionary;

determining whether the user interface context is presently being accessed by a second thread;

when the user interface context is presently being accessed by a second thread, denying the request to access the user interface context received from the first thread; and

when the user interface context is not presently being accessed by a second thread, performing a processes to allow for backward compatibility comprising:

(a) allowing the request to access the user interface context received from the first thread;

(b) updating a context record maintained by the first thread to reflect that access is allowed to the user interface context;

(c) verifying that the first thread has obtained exclusive access to the user interface context by checking the context record by way of a process comprising:

(i) incident to the first thread accessing an object in the user interface context, checking a most recent entry in the context record provided within the first thread, wherein the context record identifies the contexts accessed by the first thread, and wherein the most recent entry in the context record indicates the context that is presently being accessed by the first thread;

(ii) determining whether the most recent entry in the context record matches the user interface context associated with the object being presently accessed; and

(iii) when the most recent entry in the context record does not match the user interface context associated with the object being accessed, raising an exception; and

(d) temporarily assigning to the first thread the context settings and the context dictionary maintained by the user interface context while the first thread is operating within the user interface context, wherein assigning comprises placing the context settings and the context dictionary within thread settings of the first thread upon accessing the user interface context, and wherein settings of the context settings and dictionary information of the context dictionary are specified at a context level, rather than on a thread level.

2. (Canceled).

3. (Canceled).

4. (Previously Presented) The method for allocating the access of threads to a user interface context of claim 13, the method further comprising restoring the thread settings when a thread departs the user interface context.

5-12. (Canceled).

13. (Currently Amended) One or more computer-storage media having computer-executable instructions embodied thereon that, when executed, perform a method for allocating the access of threads to a user interface context in a threaded computing environment having a plurality of contexts, each context capable of containing a queue, context settings, a context dictionary, and objects, the method for allocating the access of threads to a user interface context comprising:

receiving a request to access the user interface context from a first thread, wherein the user interface context comprises one or more objects, wherein the user interface context is configured to receive input from a user, to provide output to the user, and to maintain the context settings and the context dictionary, wherein the context dictionary includes information from a plurality of sources;

determining whether the user interface context is presently being accessed by a second thread;

when the user interface context is presently being accessed by a second thread, denying the request to access the user interface context received from the first thread; and

when the user interface context is not presently being accessed by a second thread, performing a processes to allow for backward compatibility comprising:

(a) allowing the request to access the user interface context received from the first thread;

(b) updating a context record maintained by the first thread to reflect that access is allowed to the user interface context;

(c) verifying that the first thread has obtained exclusive access to the user interface context by checking the context record by way of a process comprising:

(i) incident to the first thread accessing an object in the user interface context, checking a most recent entry in the context

record provided within the first thread, wherein the context record identifies the contexts accessed by the first thread, and wherein the most recent entry in the context record indicates the context that is presently being accessed by the first thread;

(ii) determining whether the most recent entry in the context record matches the user interface context associated with the object being presently accessed; and

(iii) when the most recent entry in the context record does not match the user interface context associated with the object being accessed, raising an exception; and

(d) temporarily assigning to the first thread the context settings and the context dictionary maintained by the user interface context while the first thread is operating within the user interface context, wherein assigning comprises placing the context settings and the context dictionary within thread settings of the first thread upon accessing the user interface context, and wherein settings of the context settings and dictionary information of the context dictionary are specified at a context level, rather than on a thread level.

14. (Canceled).

15. (Canceled).

16. (Previously Presented) The one or more computer-storage media of claim 13, the method for allocating the access of threads to a user interface further comprising restoring the thread settings when a thread departs the user interface context.

17-24. (Canceled).